27
25 to NTIS

UCID - 16692

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the laboratory.

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore, California

TRANSIENT GAS FLOW THROUGH LAYERED POROUS MEDIA

Frank A. Morrison, Jr.

January 16, 1975

Prepared for U.S. Atomic Energy Commission under contract no. W-7405-Eng-48

TRANSIENT GAS FLOW THROUGH LAYERED POROUS MEDIA

Frank A. Morrison, Jr.

January 16, 1975

TRANSIENT GAS FLOW THROUGH LAYERED POROUS MEDIA

SUMMARY

Low Reynolds number isothermal flow of an ideal gas through layered porous material is investigated analytically. Relations governing the transient flow in one dimension are obtained. An implicit, iterative, unconditionally stable finite difference scheme is developed for calculation of such flows. A computer code, SIROCCO, employing this technique has been written and implemented on the LLL computer system. A listing of the code is included. This code may be effectively applied to the evaluation of stemming plans for underground nuclear experiments.

INTRODUCTION

Current stemming practice utilizes successive layers of different stemming materials in the column above an underground nuclear test. Any gas flow through this bed will encounter regions of widely varying permeability. An accurate description of such a flow must explicitly account for these variations. Many useful results, of course, may be obtained from the simpler analysis of uniform beds. A rapid simple method of calculating flow through actual proposed stemming configurations would, however, be particularly valuable.

Accordingly, a finite difference method for generating these results is presented here.

GOVERNING RELATION

The apparent velocity of a fluid flowing through a porous bed is, by definition, the volume flow rate per unit area normal to the direction of flow. In a low Reynolds number one-dimensional flow, this velocity, u, is given by Darcy's law

$$u = -\frac{k}{\mu}\frac{\partial p}{\partial x} \qquad (1)$$

k is the permeability of the medium, μ is the fluid viscosity, p is the fluid pressure and x is the position coordinate in the direction of flow. Because of the spatial variation of bed properties, the permeability is here a function of position.

In an incompressible porous matrix, the continuity equation expressing conservation of fluid mass may be written

$$\varepsilon \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0 \qquad (2)$$

ρ is the fluid density and ε , the porosity of the medium. The porosity is permitted to vary with position but not with time.

Substituting (1) into (2) and noting that the pressure of an isothermal ideal gas is proportional to its density, we obtain

$$\frac{\partial}{\partial x}\left(kp \frac{\partial p}{\partial x}\right) = \varepsilon\mu \frac{\partial p}{\partial t} \qquad (3)$$

This relation, with pressure as the sole dependent variable, governs the isothermal low Reynolds number flow of an ideal gas through a nonuniform porous medium.

DIMENSIONLESS FORM

The results of numerical calculations are readily applicable to a broad range of physical parameters if the calculations are done in a dimensionless form. For compatibility with other analyses and computer codes at LLL, the dimensionless groups are chosen to correspond to those of Morrison [1], Bowman [2] and Pitts [3].

Accordingly, constant reference permeability, k_o , and reference porosity, ε_o , are selected and we define

$$\bar{k} = \frac{k}{k_o} \qquad \bar{\varepsilon} = \frac{\varepsilon}{\varepsilon_o} \qquad (4)$$

We also define dimensionless pressure,

$$P = \frac{p - p_o}{p_1 - p_o} \qquad (5)$$

in terms of ambient pressure, p_o , and some larger reference pressure, p_1. When applied to underground nuclear tests, p_1 is normally taken to be the cavity pressure after cavity growth ceases. A dimensionless position is expressed as the fraction of the bed length, L , from the inlet.

$$X = \frac{x}{L} \tag{6}$$

Dimensionless time, in terms of these same quantities, is

$$\tau = \frac{k_o \, (p_1 - p_o) \, t}{\mu \, \varepsilon_o \, L^2} \tag{7}$$

A pressure ratio

$$N = \frac{p_1}{p_o} \tag{8}$$

is also defined for convenience.

In terms of these dimensionless variables, the governing equation (3) becomes

$$\frac{\partial}{\partial X} \left[\left(P + \frac{1}{N-1}\right) \bar{k} \frac{\partial P}{\partial X} \right] = \bar{\varepsilon} \frac{\partial P}{\partial \tau} \tag{9}$$

NUMERICAL TECHNIQUE

Because the relation (9) describing the flow is a nonlinear partial differential equation, its solution is best obtained by the use of numerical techniques. Because stability considerations, particularly vexatious for non-uniform beds, may severely restrict the allowable time step in an explicit calculation, an unconditionally stable implicit method was developed.

Before developing finite difference expressions, it is helpful to write (9) as

$$\frac{\partial}{\partial X} \left[\frac{\bar{k}}{2} \frac{\partial}{\partial X} \left(P^2 + \frac{2P}{N-1}\right) \right] = \bar{\varepsilon} \frac{\partial P}{\partial \tau} \tag{10}$$

and to express the dependent variable on the left side as

$$PP \equiv P^2 + \frac{2P}{N-1} \tag{11}$$

Suitable finite difference expressions may now be developed. The uniform mesh employed is shown in Figure 1. The node spacing in the X direction is ΔX and the spacial index is i . The temporal step size is $\Delta\tau$ and the time level is denoted by the index k . This index, k , will appear in the analysis only as a superscript and should not be confused with permeability.

The spatial derivative on the left side of (10) is approximated by a 3 point finite difference expression

$$\frac{\partial}{\partial X}\left[\frac{\bar{k}}{2}\frac{\partial}{\partial X}\left(P^2 + \frac{2P}{N-1}\right)\right] \sim$$
$$\frac{1}{2(\Delta X)^2}\left[\bar{k}_i\left(PP_{i+1} - PP_i\right) - \bar{k}_{i-1}\left(PP_i - PP_{i-1}\right)\right] \tag{12}$$

$\bar{k}_i$ should be interpreted as the dimensionless permeability midway between nodes i and i+1. A 3 point finite difference expression is employed because it produces a tridiagonal coefficient matrix. This matrix may be inverted quickly and efficiently.

The finite difference approximation of (10) is formed using a forward difference in time and expressing the spatial derivative as a weighted average of the finite difference expressions at the k and k+1 levels.

$$\bar{\varepsilon}_i \frac{P_i^{k+1} - P_i^k}{\Delta\tau} =$$
$$\frac{1}{2(\Delta X)^2}\left\{\beta\left[\bar{k}_i\left(PP_{i+1}^{k+1} - PP_i^{k+1}\right) - \bar{k}_{i-1}\left(PP_i^{k+1} - PP_{i-1}^{k+1}\right)\right]\right. \tag{13}$$
$$\left. + (1-\beta)\left[\bar{k}_i\left(PP_{i+1}^{k} - PP_i^{k}\right) - \bar{k}_{i-1}\left(PP_i^{k} - PP_{i-1}^{k}\right)\right]\right\}$$

β is the weighting factor. It can assume values between 0 and 1. β equal to zero corresponds to an explicit formulation. β greater than zero is implicit.

β greater than or equal to one half is stable. β equal to one half has been satisfactory in all calculations thus far, but the option of increasing β is retained should oscillations appear in any results.

We now define

$$h = \frac{(\Delta X)^2}{\Delta\tau} \tag{14}$$

and rearrange (13) with unknowns, variables at the k+1 time level, on the left and known quantities, those evaluated at the k time level, on the right.

$$\begin{aligned} &4h\,\bar{\varepsilon}_i\,P_i^{k+1} - 2\beta\left[\bar{k}_i\left(PP_{i+1}^{k+1} - PP_i^{k+1}\right)\right. \\ &\left. - \bar{k}_{i-1}\left(PP_i^{k+1} - PP_{i-1}^{k+1}\right)\right] = 4h\,\bar{\varepsilon}_i\,P_i^{\,k} \\ &+ 2(1-\beta)\left[\bar{k}_i\left(PP_{i+1}^k - PP_i^k\right) - \bar{k}_{i-1}\left(PP_i^k - PP_{i-1}^k\right)\right] \end{aligned} \tag{15}$$

For compactness of notation, the known expression at the k time level is denoted by D_i

$$D_i \equiv 4h\,\bar{\varepsilon}_i\,P_i^k + 2(1-\beta)\left[\bar{k}_i\left(PP_{i+1}^k - PP_i^k\right) - \bar{k}_{i-1}\left(PP_i^k - PP_{i-1}^k\right)\right] \tag{16}$$

If we simplify further by dropping the k+1 superscript for P and PP at the k+1 level, (15) becomes

$$4h\,\bar{\varepsilon}_i\,P_i - 2\beta\left[\bar{k}_i\left(PP_{i+1} - PP_i\right) - \bar{k}_{i-1}\left(PP_i - PP_{i-1}\right)\right] = D_i \tag{17}$$

Because the equation (17) is nonlinear, it is solved iteratively. The nonlinear portion is factored and then linearized by assuming a value for a portion of the expression. Because the assumed value may not be presumed correct, an iterative procedure is employed to generate progressively better assumptions. Using the definition (11), we find

$$PP_{i+1} - PP_i = \left(P_{i+1} + P_i + \frac{2}{N-1}\right)\left(P_{i+1} - P_i\right) \tag{18}$$

and

$$PP_i - PP_{i-1} = \left(P_i + P_{i-1} + \frac{2}{N-1}\right)\left(P_i - P_{i-1}\right) \tag{19}$$

The expressions are linearized by replacing the first factor by the equivalent expression in terms of assumed pressures, denoted by AP, at the new time level.

$$\begin{aligned} 4h\,\bar{\varepsilon}_i\,P_i &- 2\beta\,\bar{k}_i\left(AP_{i+1} + AP_i + \frac{2}{N-1}\right)\left(P_{i+1} - P_i\right) \\ &+ 2\beta\,\bar{k}_{i-1}\left(AP_i + AP_{i-1} + \frac{2}{N-1}\right)\left(P_i - P_{i-1}\right) = D_i \end{aligned} \tag{20}$$

We may write this relation among the unknown P's as

$$A_i P_{i-1} + B_i P_i + C_i P_{i+1} = D_i \tag{21}$$

where

$$\begin{aligned} A_i &= -2\beta\,\bar{k}_{i-1}\left(AP_i + AP_{i-1} + \frac{2}{N-1}\right) \\ B_i &= 4h\,\bar{\varepsilon}_i + 2\beta\left[\bar{k}_i\left(AP_{i+1} + AP_i + \frac{2}{N-1}\right) + \bar{k}_{i-1}\left(AP_i + AP_{i-1} + \frac{2}{N-1}\right)\right] \\ C_i &= -2\beta\,\bar{k}_i\left(AP_{i+1} + AP_i + \frac{2}{N-1}\right) \end{aligned} \tag{22}$$

If β is chosen to be one half and if we consider only uniform media with $\bar{\varepsilon}$ and $\bar{k}$ equal to one, this formulation reduces to that of Bruce, Peaceman, Rachford and Rice [4]. The procedure of Bruce, Peaceman, Rachford and Rice is employed in the DIASPORA code [5] describing convective-dispersive transport in porous media.

With coefficients A_i, B_i, C_i, D_i, the tridiagonal system of equations (21) is solved, using Thomas' algorithm [6], to yield an estimate of each P_i at time level k+1. This estimate is then used as the next assumed pressure, AP_i, and the process repeated until some convergence criterion is satisfied. The criterion used in the SIROCCO code is

$$\sum_i \left| P_i - AP_i \right| \leq 10^{-4} \tag{23}$$

The first set of assumed pressures at any time level are generated using an explicit calculation. The explicit formulation results from setting β equal to zero in equations (16) and (22). Following the explicit calculation, at least one implicit calculation is done. Each implicit calculation is followed by the convergence test.

BOUNDARY CONDITIONS

The pressure at the bed inlet, X equal to zero, is specified. At the other end of the bed, two conditions are of particular interest. If the bed is open to the atmosphere and fluctuations in barometric pressure are neglected, the dimensionless pressure P_n at the exit remains equal to zero. Substitution of the relations

$$A_n = D_n = 0 \quad , \quad B_n = 1 \tag{24}$$

into the governing set of equations yields the desired result.

If the bed is sealed, then no flow crosses the far boundary. From (1), it follows that the pressure gradient vanishes there. This condition can be approximated by considering a phantom node, denoted by n + 1, located a distance ΔX beyond the boundary. The pressure at this phantom node is taken to be the same as the pressure an equal distance inside the boundary

$$P_{n+1} = P_{n-1} \tag{25}$$

and the permeability is taken to be uniform in this region

$$\bar{k}_i = \bar{k}_{i-1} \tag{26}$$

Writing (21) for the boundary, i equal to n , substituting (25) and (26) and combining coefficients, there results in place of (16) and (22)

$$A_n = - 4\beta \bar{k}_{n-1} \left(AP_n + AP_{n-1} + \frac{2}{N-1} \right)$$

$$B_n = 4h \bar{\varepsilon}_n + 4\beta \bar{k}_{n-1} \left(AP_n + AP_{n-1} + \frac{2}{N-1} \right) \tag{27}$$

$$D_n = 4h \bar{\varepsilon}_n P_n + 2(1-\beta) \bar{k}_{n-1} \left(PP_{n-1} - PP_n \right)$$

The explicit expression, needed to generate the first boundary pressure estimate at each time level, is obtained by setting β equal to zero in (27).

DISCUSSION OF COMPUTER CODE SIROCCO

The analysis presented has been incorporated in a computer code SIROCCO and implemented on the LLL CDC 7600s. The code has been tested. Results for uniform beds agree with calculations of previous codes. Results for non-uniform beds appear well behaved and consistent with anticipated behavior. A listing of this code appears as an appendix.

The user must specify:

1. number of nodes
2. time step. This time step may be changed once during execution at a user selected time.
3. pressure ratio, N.
4. number and times of pressure versus position plots
5. number and times of printed pressure distribution
6. uniform or non-uniform porosity
7. uniform or non-uniform permeability
8. uniform or non-uniform initial pressure distribution

9. weighting factor for implicit pressure calculation
10. open or closed column.

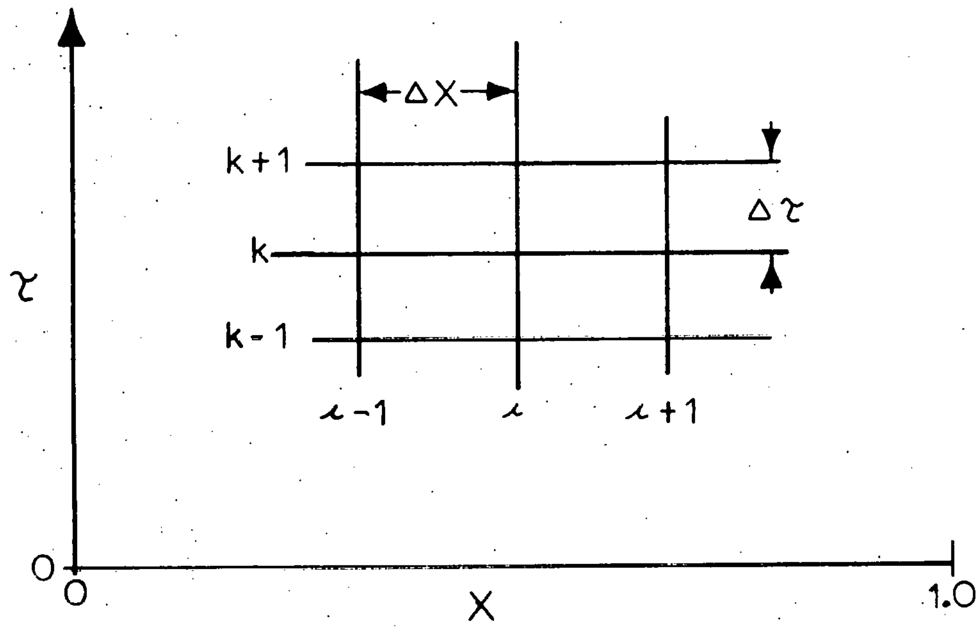

FIG. 1 - MESH FOR FINITE DIFFERENCE CALCULATIONS

REFERENCES

1. Morrison, F. A. Jr., "Transient Gas Flow in a Porous Column," I&EC Fundamentals, 11, 191-197, 1972.

2. Bowman, B. R., "TWIG, A Computer Code for the Calculation of Two-Dimensional, Ideal Gas Flow in Porous Media," UCID-16320, Lawrence Livermore Laboratory, 1972.

3. Pitts, J. H., "Gas-Initiated Crack Propagation in a Porous Solid - A Progress Report," UCID-16236, Lawrence Livermore Laboratory, 1973.

4. Bruce, G. H., Peaceman, D. W., Rachford, H. H. and Rice, J. D., "Calculations of Unsteady-State Gas Flow through Porous Media," Pet. Trans. AIME 198, 79-92, 1953.

5. Morrison, F. A., Jr., "An Eulerian-Lagrangian Method for Convective-Dispersive Transport" presented at CUBE Symposium, Oct. 23-25, 1974, Lawrence Livermore Laboratory.

6. von Rosenberg, D. U., "Methods for the Numerical Solution of Partial Differential Equations," American Elsevier, New York, 1969.

```
*ID    751NTT                                              MORRISON          BOX R04
* CONTROLLEE 0 +FRANK
* DROP +FRANKD
* LIST8
* CARDS DBUG DIMP
* FORTRAN
      PROGRAM FAM(TAPE59,OUTPUT,TAPE3=OUTPUT,INPUT,TAPE2=INPUT)
C
C
C                             SIROCCO
C
C
C     LOW REYNOLDS NUMBER ISOTHERMAL IDEAL GAS FLOW IN
C     POROUS MEDIA IS CALCULATED. AN IMPLICIT FINITE
C     DIFFERENCE METHOD IS USED. SPATIAL VARIATIONS OF
C     PERMEABILITY AND POROSITY ARE PERMITTED. PRESSURE
C     IS SPECIFIED AT THE INLET. THE OUTLET PRESSURE IS
C     SPECIFIED OR, ALTERNATIVELY, THE OUTLET IS SEALED.
C
C     ................NOMENCLATURE.....................
C
C     I      -EULERIAN NODE INDEX
C     TAU    -DIMENSIONLESS TIME
C     X      -DIMENSIONLESS POSITION
C     DELTAU-DIMENSIONLESS TIME INCREMENT
C     DX     -DIMENSIONLESS EULERIAN POSITION INCREMENT
C     H      -DX*DX/DELTAU
C     N      -PRESSURE RATIO
C     P(I)   -DIMENSIONLESS PRESSURE AT EULERIAN NODE I
C     PP(I)  -P*P+2P/(N-1)
C     AP(I)  -DIMENSIONLESS ASSUMED PRESSURE
C     NEULER-NUMBER OF EULERIAN NODES
C     DETAU1-PRIMARY DELTAU
C     TAUDT  -APPROXIMATE TIME OF CONSTANT TIME STEP
C     IFACTR-INITIAL REDUCTION OF DELTAU
C     DTCON  -CONSTANT TIME STEP CONTROL (LOGICAL)
C     ITER   -ITERATION COUNTER FOR NUMBER OF TIME STEPS
C     NITER  -ITERATION NUMBER IN PRESSURE SOLVING ROUTINE
C     PLTIME-ARRAY OF DIMENSIONLESS TIMES FOR PLOTS
C     NPLOT  -NUMBER OF PLOTS PER RUN
C     NPRINT-NUMBER OF TIMES EULERIAN OUTPUT PRINTED
C     PRTIME-ARRAY OF DIMENSIONLESS TIMES FOR PRINTS
C     EPSLON-DIMENSIONLESS POROSITY @  NODE I
C     PERM   -DIMENSIONLESS PERMEABILITY AT I+1/2
C     BETA   -WEIGHTING FACTOR FOR IMPLICIT PRESSURE CALCULATION
C     OPEN   -OUTLET BOUNDARY CONTROL (LOGICAL)
C
      CALL DEVICE(6HCREATE,6HOUTPUT,500000)
      DIMENSION AA(102),BB(102),CC(102),DD(102)
      DIMENSION PLTIME(10),PRTIME(100)
      DIMENSION P(102),AP(102),PP(102),X(102),CONVRG(10)
      DIMENSION EPSLON(102),PERM(102)
      COMMON/SOLV/AA,BB,CC,DD
      COMMON/EULE/P,CONVRG,NEULER,NITER
      COMMON/TIME/TAU,ITER
      COMMON/GRAF/N,PLTIME,NPLOT,OPEN
      INTEGER TITLE,EPRT
      INTEGER PLTCHK,EULCHK,PORCHK,PRMCHK,PINCHK
      LOGICAL DTCON,CHANGE,OPEN,PORCON,PRMCON,PINCON
      LOGICAL PLOTED,NOPLOT,NOSPKE
```

```
      REAL N,NN
      NAMELIST/HEAR/NEULER,DETAU1,TAUDT,IFACTR,N,
     1 NPLOT,NPRINT,PORCON,PRMCON,PINCON,DTCON,BETA,OPEN
C
C        READ INPUT
C
      DATA NMLT/"NAME"/
      DATA PLTCHK/"PLOT"/
      DATA EULCHK/"PRIN"/
      DATA PORCHK/"PORO"/
      DATA PRMCHK/"PERM"/
      DATA PINCHK/"INIT"/
      READ(2,0) TITLE
      WRITE(3,0) TITLE
    0 FORMAT(1A5)
      CALL DD80ID (TITLE,1)
    1 CONTINUE
      READ(2,10) NAM
      WRITE(3,10) NAM
   10 FORMAT(1A4)
      IF(NAM.NE.NMLT) GO TO 130
      INPUT DATA HEAR,2,3
      NOPLOT=.FALSE.
      NOSPKE=.FALSE.
      IF(NPLOT.EQ.0) NOPLOT=.TRUE.
      IF(NPRINT.EQ.0) NOSPKE=.TRUE.
      NEMIN1=NEULER-1
      READ(2,10) NAM
      WRITE(3,10) NAM
      IF(NAM.NE.PLTCHK) GO TO 125
      IF(NOPLOT) READ(2,10) NAM
      IF(NOPLOT) WRITE(3,10) NAM
      IF(.NOT.NOPLOT) READ(2,15) (PLTIME(I), I=1,NPLOT)
      IF(.NOT.NOPLOT) WRITE(3,15) (PLTIME(I), I=1,NPLOT)
   15 FORMAT(8E10.3)
      READ(2,10) NAM
      WRITE(3,10) NAM
      IF(NAM.NE.EULCHK) GO TO 125
      IF(NOSPKE) READ(2,10) NAM
      IF(NOSPKE) WRITE(3,10) NAM
      IF(.NOT.NOSPKE) READ(2,15) (PRTIME(I), I=1,NPRINT)
      IF(.NOT.NOSPKE) WRITE(3,15) (PRTIME(I), I=1,NPRINT)
      READ(2,10) NAM
      WRITE(3,10) NAM
      IF(NAM.NE.PORCHK) GO TO 125
      IF(PORCON) GO TO 16
      READ(2,15) (EPSLON(I), I=1,NEULER)
      WRITE(3,15) (EPSLON(I), I=1,NEULER)
      GO TO 17
   16 READ(2,10) NAM
      WRITE(3,10) NAM
      DO 17 I=1,NEULER
      EPSLON(I)=1.
   17 CONTINUE
      READ(2,10) NAM
      WRITE(3,10) NAM
      IF(NAM.NE.PRMCHK) GO TO 125
      IF(PRMCON) GO TO 18
      READ(2,15) (PERM(I), I=1,NEMIN1)
      WRITE(3,15) (PERM(I), I=1,NEMIN1)
```

```
          GO TO 19
   18     READ(2,10) NAM
          WRITE(3,10) NAM
          DO 19 I=1,NEMIN1
          PERM(I)=1.
   19     CONTINUE
          READ(2,10) NAM
          WRITE(3,10) NAM
          IF(NAM.NE.PINCHK) GO TO 125
          IF(PINCON) GO TO 20
          READ(2,15) (P(I), I=1,NEULER)
          WRITE(3,15) (P(I), I=1,NEULER)
          GO TO 21
   20     READ(2,10) NAM
          WRITE(3,10) NAM
          P(1)=1.
          DO 21 I=2,NEULER
          P(I)=0.
   21     CONTINUE
          CALL DRAW
C
C                SET SYSTEM PARAMETERS
C
          CHANGE=.NOT.DTCON
          KPLOT=1
          EPRT=1
          ICHECK=0
          NN=2./(N-1.)
          DX=1./(FLOAT(NEULER)-1.)
          TAU=0.
          ITER=0
          NITER=1
          TEST=.0001
          DELTAU=DETAU1
          IF(.NOT.DTCON) DELTAU=DETAU1/FLOAT(IFACTR)
          H=DX*DX/DELTAU
          CONVRG(1)=0.0
C
C                             SET X ARRAY
C
          DO 30 I=1,NEULER
          RI=I
   30     X(I)=(RI-1.)*DX
          AP(1)=1.
          IF(OPEN) AP(NEULER)=0.
   51     IF(.NOT.NOSPKE) CALL SPEAK
C
C                        ITERATIONS ON TIME
C
   55     CONTINUE
          ITER=ITER+1
          TAU=TAU+DELTAU
   57     IF(TAU.GE.TAUDT-DELTAU/2.) DTCON=.TRUE.
C
C                                     FIND NEW PRESSURES
C           FINDING THE FIRST ASSUMED P'S USING THE EXPLICIT METHOD
C
          DO 60 I=1,NEULER
   60     PP(I)=P(I)*(P(I)+NN)
          DO 65 I=2,NEMIN1
```

```
      AP(I)=P(I)+(PERM(I)*(PP(I+1)-PP(I))-PERM(I-1)*
     1 (PP(I)-PP(I-1)))/(2.*H*EPSLON(I))
   65 CONTINUE
      IF(.NOT.OPEN) AP(NEULER)=P(NEULER)+PERM(NEULER-1)*
     1 (PP(NEULER-1)-PP(NEULER))/(H*EPSLON(NEULER))
C
C     SOLVING FOR THE NEW PRESSURES
C      SET UP ARRAY OF CONSTANTS ON R.H.S.--DD'S
C
      DD(1)=1.
      IF(OPEN) DD(NEULER)=0.
      IF(.NOT.OPEN) DD(NEULER)=4.*H*EPSLON(NEULER)*P(NEULER)
     1 +2.*(1.-BETA)*PERM(NEULER-1)*(PP(NEULER-1)-PP(NEULER))
      DO 66 I=2,NEMIN1
      DD(I)=4.*H*EPSLON(I)*P(I)+2.*(1.-BETA)*(PERM(I)*
     1 (PP(I+1)-PP(I))-PERM(I-1)*(PP(I)-PP(I-1)))
   66 CONTINUE
C
C      SET UP COEFFICIENT ARRAYS--AA'S,BB'S,CC'S
C
      BB(1)=1.
      CC(1)=0.
      IF(OPEN) AA(NEULER)=0.
      IF(OPEN) BB(NEULER)=1.
C
C         ITERATIONS SOLVING FOR P'S.
C
      DO 73 NITER=1,10
      DO 71 I=2,NEMIN1
      AA(I)=-2.*BETA*PERM(I-1)*(AP(I)+AP(I-1)+NN)
      BB(I)=4.*H*EPSLON(I)+2.*BETA*(PERM(I)*
     1 (AP(I+1)+AP(I)+NN)+PERM(I-1)*(AP(I)+AP(I-1)+NN))
      CC(I)=-2.*BETA*PERM(I)*(AP(I+1)+AP(I)+NN)
   71 CONTINUE
      IF(.NOT.OPEN) AA(NEULER)=-4.*BETA*PERM(NEULER-1)*
     1 (AP(NEULER)+AP(NEULER-1)+NN)
      IF(.NOT.OPEN) BB(NEULER)=4.*H*EPSLON(NEULER)
     1 +4.*BETA*PERM(NEULER-1)*(AP(NEULER)+AP(NEULER-1)+NN)
      CALL SOLVER(P,1,NEULER)
C
C            CALCULATE CONVERGENCE INDICATORS
C
      CONVRG(NITER)=0.0
      DO 72 I=2,NEULER
      CONVRG(NITER)=ABS(P(I)-AP(I))+CONVRG(NITER)
   72 AP(I)=P(I)
C
C            TEST FOR CONVERGENCE OF PRESSURES
C
      IF(CONVRG(NITER).LE.TEST) GO TO 74
   73 CONTINUE
   74 CONTINUE
C
C            TIME TO PLOT?
C
      PLOTED=.FALSE.
      IF(NOPLOT) GO TO 116
      IF(TAU+DELTAU/2.-PLTIME(KPLOT)) 116,114,114
  114 IF(KPLOT.EQ.NPLOT)  NOPLOT=.TRUE.
      PRINT 115, TAU,ITER
```

```
  115 FORMAT(" ",///," AN ATTEMPT WAS MADE TO PLOT AT TAU="
     1 ,F10.6,"  ITER=",I8,//)
      CALL TRACE(X,P,NEULER)
      KPLOT=KPLOT+1
      PLOTED=.TRUE.
      CALL SPEAK
  116 CONTINUE
C
C               TIME FOR SPEAK?
C
  118 IF(NOSPKE) GO TO 120
      IF(TAU+DELTAU/2.-PRTIME(EPRT)) 120,119,119
  119 IF(EPRT.EQ.NPRINT) NOSPKE=.TRUE.
      EPRT=EPRT+1
      IF(.NOT.PLOTED) CALL SPEAK
  120 IF(NOPLOT.AND.NOSPKE) GO TO 1
C
C               CHANGE TIME STEP?
C
      IF(.NOT.CHANGE) GO TO 55
      ICHECK=ICHECK+1
      IF(ICHECK.NE.IFACTR) GO TO 55
      ICHECK=0
      IF(.NOT.DTCON) GO TO 55
      DELTAU=DETAU1
      H=DX*DX/DELTAU
      CHANGE=.FALSE.
      GO TO 55
  125 CONTINUE
      WRITE(3,126)
  126 FORMAT(" HEADING CHECK ERROR")
  130 CONTINUE
      CALL KEEP80(1)
      CALL PLOTEA
      CALL EXIT(1)
      END
      SUBROUTINE SOLVER(X,IFIRST,ILAST)
C
C     THIS SUBROUTINE SOLVES A SET OF LINEAR EQUATIONS, WHERE THE
C     COEFFICIENT ARRAY IS TRI-DIAGONAL.
C     THE EQUATIONS ARE NUMBERED FROM IFIRST TO ILAST AND THEIR SUB-DIAGONAL,
C     DIAGONAL, AND SUPER DIAGONAL COEFFICIENTS ARE STORED IN ARRAYS A,
C     B, AND C.  THE RIGHT HAND SIDE CONSTANTS ARE IN ARRAY D.  THE
C     SOLUTION IS ARRAY X.
C
      DIMENSION A(102),B(102),C(102),D(102),X(ILAST)
      DIMENSION G(102),W(102),BA(102)
      COMMON/SOLV/A,B,C,D
      W(IFIRST)=B(IFIRST)
      G(IFIRST)=D(IFIRST)/W(IFIPST)
      IFPLUS=IFIRST+1
      DO 1 I=IFPLUS,ILAST
      IM1=I-1
      BA(IM1)=C(IM1)/W(IM1)
[illegible]
    1 G(I)=(D(I)-A(I)*G(IM1))/W(I)
      X(ILAST)=G(ILAST)
      J=ILAST
      DO 2 I=IFPLUS,ILAST
      J=J-1
```

```
    2   X(J)=G(J)-BA(J)*X(J+1)
        RETURN
        END
        SUBROUTINE SPEAK
C
C       THIS SUBROUTINE PRINTS OUT THE PRESSURE INFORMATION.
C
        DIMENSION P(102),E(10)
        COMMON/EULE/P,E,NEULER,NITER
        COMMON/TIME/TAU,ITER
        PRINT 10,TAU,ITER
        PRINT 11
        PRINT 12,(P(M), M=1,NEULER)
        PRINT 13,NITER
        PRINT 14,(E(M), M=1,NITER)
   10   FORMAT(/,11H TIME TAU= ,F9.7," NO. OF TIME STEPS= ",I10)
   11   FORMAT( 10H THE PRESSURES ARE )
   12   FORMAT( 11F9.5)
   13   FORMAT(/," NUMBER OF ITERATIONS = ",I3,
       1 ", CONVERGENCE INDICATORS ARE")
   14   FORMAT(" ",10F9.6)
        RETURN
        END
        SUBROUTINE DRAW
C
C       THIS SUBROUTINE PREPARES THE FRAME
C
        DIMENSION PLTIME(10)
        COMMON/GRAF/N,PLTIME,NPLOT,OPEN
        CALL FRAME
        CALL MAPS(0.,1.,0.,1.,.1,.9,.1,.9)
        CALL SETCH(60.,3.,1,0,0,0,0)
        WOT 100,10
   10   FORMAT("POSITION X")
        CALL SETCH(26.,60.,1,0,0,0,0)
        WOT 100,15,(PLTIME(I),I=1,NPLOT)
   15   FORMAT("PLOT TIMES =",10F8.3)
        CALL SETCH(3.,57.,1,0,0,1,0)
   40   FORMAT("  PRESSURE P")
        CALL SETCH(53.,64.,1,0,0,0,0)
        WOT 100,45,N
   45   FORMAT("PRESSURE RATIO =",F5.1)
        CALL SETCH(45.,62.,1,0,0,0,0)
        IF(OPEN) WOT 100,50
        IF(.NOT.OPEN) WOT 100,51
   50   FORMAT("          OPEN COLUMN")
   51   FORMAT("         SEALED COLUMN")
        RETURN
        END
```

FAM:nf

Interal Distribution :

W. Arnold, L-123
A. Austin, L-505
B. Bowman, L-90
H. Brandt, L-123
J. Carothers, L-07
R. Gelinas, L-71
L. Germain, L-7
J. Grens, L-437
J. Hadley, L-531
R. Heckman, L-231
B. Hudson, L-48
J. Kahn, L-41
V. Karpenko, L-113
V. Kopytoff, L-90
R. Martin, L-90
E. McCauley, L-90
W. McMaster, L-42
J. Pitts, L-90 10 copies
R. Rozsa, L-437
J. Shearer, L-18
D. Snoeberger, L-437
S. Sutton, L-90
F. Tokarz, L-90
J. Toman, L-47

TID 15 copies
TIC 27 copies

External Distribution :

Professor Frank A. Morrison, Jr.
Mechanical & Industrial Engineering Department
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy \$___*; Microfiche \$2.25

*Pages	NTIS Selling Price
1-50	\$4.00
51-150	\$5.45
151-325	\$7.60
326-500	\$10.60
501-1000	\$13.60